REMARKS/ARGUMENTS

Claims 1-20 are pending in the present application. Claim 7 has been amended. Support for the amendment can be found in the specification on page 21, lines 9-29. Reconsideration of the claims is respectfully requested.

I. Interview Summary

Applicants would like to thank Examiner Gelagay for conducting a telephone interview on August 13, 2007, in regards to the §112 and § 101 rejections of claim 7. An agreement was reached as to possible amendments that would overcome the § 112 and § 101 rejections.

In addition, Applicants discussed the § 103 rejection of claim 1. In particular, Applicants discussed with the Examiner that the prior art references did not teach several features of claim 1. The Examiner considered how the prior art references failed to teach certain features recited in claim 1. However, no agreement was reached.

II. 35 U.S.C. § 112, Second Paragraph

The Examiner rejects claims 7-9 under 35 U.S.C. § 112, second paragraph, as indefinite. This rejection is respectfully traversed. The Examiner asserts the following:

> Claims 7 provides for the use of "a second value", but, since the claim does not set forth any steps involved in the method/process, it is unclear what method/process applicant is intending to encompass. A claim is indefinite where it merely recites a use without any active, positive steps delimiting how this use is actually practiced.
>
> Claim 7 is rejected under 35 U.S.C. 101 because the claimed recitation of a use, without setting forth any steps involved in the process, results in an improper definition of a process, i.e., results in a claim which is not a proper process claim under 35 U.S.C. 101. See for example *Ex parte Dunki*, 153 USPQ 678 (Bd.App. 1967) and *Clinical Products, Ltd.* v. *Brenner*, 255 F. Supp. 131, 149 USPQ 475 (D.D.C. 1966).

Office Action dated May 15, 2007, page 2.

Applicants have amended claim 7 accordingly to set forth the positive steps of the process. Therefore, the rejection of claim 7 under 35 U.S.C. § 112, second paragraph has been overcome.

III. 35 U.S.C. § 101

The Examiner rejects claims 7-9 under 35 U.S.C. § 101 as directed towards non-statutory subject matter. Applicants have amended claim 7 accordingly to produce a useful, concrete, and tangible result. Thus, the rejection of claims 7-9 under 35 U.S.C. § 101 have been overcome.

IV. 35 U.S.C. § 103, Obviousness

The Examiner rejects claims 1-20 under 35 U.S.C. § 103 as obvious over *Schneier et al.*, Event Auditing System, U.S. Patent 5,978,475, November 2, 1999, (hereinafter "*Schneier*") in view of *Grawrock, et al.*, Validation of Inclusion of a Platform Within a Data Center, U.S. Patent 7,058,807, June 6, 2006, (hereinafter "*Grawrock*"). This rejection is respectfully traversed. In regards to claims 1, 7, and 10, the Examiner asserts the following:

> As per claims 1, 7 and 10:
> Schneier teaches a method of logging audit events in a data processing system, the method comprising the computer implemented steps of:
> writing a sequence of audit records including a final audit record to a first log file stored by a data processing system; (figure 3, col. 7, lines 1-25)
> calculating a respective first hash value of each audit record; (figure 3)
> responsive to calculating each respective first hash value, calculating a corresponding second hash value from the first hash value (col. 10, lines 16-35)
> writing the second hash value to the register; (col. 10, lines 16-35)
> responsive to closing the first log file, opening a second log file; (col. 7, lines 1- 18) and
> writing, to a first record of the second log file, a final second hash value corresponding to a first hash value of the final audit record. (col. 10, line 64-col. 11, line 54)
> Schneier does not explicitly disclose calculating a corresponding second hash value from a value of a register associated with the data processing system and writing the second hash value to the register. Grawrock in analogous art, however, discloses calculating a corresponding second hash value from a value of a register associated with the data processing system and writing the second hash value to the register. (col. 5, line 46-col. 6, line 16) Therefore it would have been obvious to one ordinary skill in the art to modify the method disclosed by Schneier with Grawrock in order to have a trusted platform module that provides an increased confidence and that enables enhancement of auditing and logging (col. 1, lines 25-31; Grawrock).

Office Action dated May 15, 2007, pages 3-4.

IV.A. The Examiner Has Failed to State a *Prima Facie* Case of Obviousness Because the References Do Not Teach or Suggest All the Features of Claim 1.

Regarding claims 1, 7, and 10, the Examiner has failed to state a *prima facie* obviousness rejection because the proposed combination does not teach or suggest all of the features of the claims. A *prima facie* case of obviousness is established when the teachings of the prior art itself suggest the claimed subject matter to a person of ordinary skill in the art. *In re Bell*, 991 F.2d 781, 783, 26 U.S.P.Q.2d 1529, 1531 (Fed. Cir. 1993). If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988). All limitations of the claimed invention must be considered when determining patentability. *In re*

Lowry, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). In the case at hand, not all of the features of the claimed invention have been properly considered and the teachings of the references themselves do not teach or suggest the claimed subject matter to a person of ordinary skill in the art.

Claim 1 is representative of this grouping of claims. Claim 1 is as follows:

> 1. A method of logging audit events in a data processing system, the method comprising the computer implemented steps of:
> writing a sequence of audit records including a final audit record to a first log file stored by a data processing system;
> calculating a respective first hash value of each audit record;
> responsive to calculating each respective first hash value, calculating a corresponding second hash value from the first hash value and a value of a register associated with the data processing system;
> writing the second hash value to the register;
> responsive to closing the first log file, opening a second log file; and
> writing, to a first record of the second log file, a final second hash value corresponding to a first hash value of the final audit record.

The Examiner has failed to state a *prima facie* obviousness rejection against claim 1 because neither *Schneier* nor *Grawrock* teach or suggest all features of claim 1. For example, *Schneier* does not teach or suggest the feature "responsive to closing the first log file, opening a second log file" as recited in claim 1. The Examiner asserts otherwise, citing to the following section:

> D. Logging Operations
> 1. Overview
> The method of the invention may involve the following phases:
> Startup--An initial interaction with the trusted computer to allow the untrusted computer to create a new log file.
> Writing--As events occur and are logged, new entries are added to the log file on the untrusted computer.
> Posting--Occasionally during the course of the logging, the untrusted computer may write "summary records" into its logfile to allow independent chunks of the logfile to be independently verified and accessed independent of all the others.
> Closing--A final set of operations to seal the logfile and transmit it to the trusted computer for verification.
> Verification--The set of operations done on the logfile to guarantee that it hasn't been altered or had any records deleted by someone who didn't know the authentication key for that record.

Schneier, column 7, lines 1-18.

The above portion of *Schneier* presents an overview of the logging operations between a trusted and untrusted computer. At startup, the untrusted computer creates a new log file. New entries are added to the log file on the untrusted computer as events occur. The untrusted computer may write "summary records" into its logfile to allow independent chunks of the logfile to be independently verified. Upon closing, a final

set of operations to seal the logfile and transmit it to the trusted computer for verification occurs. A set of operations will be performed on the logfile to verify that it has not been altered.

Schneier does not teach the feature "responsive to closing the first log file, opening a second log file" as recited in claim 1. The above portion of *Schneier* does not teach upon closing the logfile, a second logfile is opened. *Schneier* teaches creating/opening a new log file only during the initial interaction with the trusted computer and the untrusted computer. *Schneier* is devoid of any teaching in regards to the opening of a second log file upon the closing of a first log file. *Schneier* also does not suggest the feature "responsive to closing the first log file, opening a second log file" as recited in claim 1.

Additionally, *Schneier* does not teach the feature "writing, to a first record of the second log file, a final second hash value corresponding to a first hash value of the final audit record," as recited in claim 1. The Examiner asserts otherwise, citing the following section of *Schneider*:

> b. Writing
>
> Once the logfile has been created and the initial secrets exchanged, the logfile is initialized for writing. FIG. 5 shows the process steps by which U adds an entry to the audit log 300. At step 510, the specific data D.sub.j to be logged in the entry is obtained (e.g., generated by a program running on the computer 102, or obtained in real-time from its input device 145, or aggregated in its storage device 160, or obtained from a remote device accessed via external interface 180). In general, the data can include raw binary data, text, program code, and any other information storable in a computer-readable format. The specific data format of D.sub.j is not restricted; it need only be something that a verifier of the log entries will unambiguously understand, and that can be distinguished virtually all cases from random gibberish. That is, the data must enable detection of an altered record by making it decrypt to an obviously incorrect answer. Structure can optionally be imposed on the data to help the verifier perform such detection. This is especially useful where the data are to be provided to a partially trusted verifier, but who is not authorized to receive the authentication key used to create the MAC that absolutely guarantees the data (as will be discussed in greater detail below). For example, where the data are CBC-mode block ciphers or standard stream ciphers, a very simple checksum at the end of the message will be sufficient. In other cases, a hash operator might be appropriate.
>
> W.sub.j is the log entry type of the j-th log entry. At step 515, the log entry type appropriate to the data is specified. The details are necessarily implementation-specific but the log entry type might typically be a fixed length (e.g., 32-bit) word having a small number of values (e.g., in the example described previously, the range from 0xffffff00 to 0xffffffff) reserved for system use, with all others being available for the application program to define as necessary. The log entry type can serve as categories of error message, and also for access control: partially-trusted users can be given or denied access to various log entries based upon their types. Thus, the log entry type is also called a permission mask. In an exemplary embodiment of the invention, permission masks are determined according to a universal, public standard. Thus, anyone

knowing the identity of a partially-trusted user can determine an appropriate permission mask.

A.sub.j =hash ("Increment Hash", A.sub.j-1) is the authentication key used in the MAC that provides cryptographically verifiable authenticity of the j-th entry in the log. The log's authentication key is hashed immediately after a log entry is written, creating a new value for the authentication key that overwrites and irretrievably deletes the previous value. The authentication key for the current log entry is therefore created at step 570 of the previous log entry. The original value, A.sub.0, is given to U by T at startup using the process described previously. Alternatively, U can generate A.sub.o and then securely transmit it to T (who will need A.sub.0 for log verification, described later). By changing authentication keys each time, and never storing past values, even if U becomes compromised the security of entries prior to the compromise remains intact.

Schneier, column 10, line 64 – column 11, line 54.

The above portion of *Schneier* discusses the process by which the untrusted computer adds an entry to the audit log. The first paragraph states that the log data can include raw binary data, text, program code, and any other information storable in a computer-readable format as long as the data is understandable to a verifier.

The second paragraph states that the log entry type appropriate to the data is specified. The log entry type might typically be a fixed length (e.g., 32-bit) word. The log entry type can serve as categories of error messages, and also for access control. Thus, the log entry type is also called a permission mask.

The third paragraph above relates to the authentication key. The authentication key is used in the message authentication codes (MAC) to provide cryptographically verifiable authenticity of the entries in the log. The log's authentication key is hashed immediately after a log entry is written, creating a new value for the authentication key that overwrites and irretrievably deletes the previous value. The authentication key for the current log entry is therefore created from the previous log entry.

The portion of *Schneier* cited above does not teach the feature "writing, to a first record of the second log file, a final second hash value corresponding to a first hash value of the final audit record" as recited in claim 1. *Schneier* does not teach writing to a first record of a second log file. The above portion discusses writing entries in the same log file, wherein the authentication key for each entry is generated by hashing the previous authentication key value. Because the entries are written to the same log file, the entries are not written to a second log file, as claimed.

Furthermore, *Schneier* does not suggest writing to the first record in a second log file, the hash value corresponding to the hash value of the final audit record in the first log file. *Schneier* actually teaches away from the claimed features of claim 1. *Schneier* teaches that a log file is created upon an initial interaction between the trusted computer and the untrusted computer (*Schneier*, column 7, lines 5-6). The initial/original authentication key value, A.sub.0, is given to the untrusted computer by the trusted computer

at startup or the untrusted computer can generate A.sub.0 and then securely transmit it to trusted computer (*Schneier*, column 11, lines 47-51). A.sub.0 is a random starting authentication key (*Schneier*, column 9, line 55). Therefore, *Schneier* teaches that upon the creation of a log file, the initial starting authentication key is a random authentication key either generated by the trusted computer or the untrusted computer.

Contrary to *Schneier*, claim 1 recites that the first record in a second log file is written to the hash value corresponding to the hash value of the final audit record in the first log file. Thus, the initial hash value corresponds to the hash value of the final audit record in the first log file, whereas, *Schneier* teaches starting with a random authentication key. Accordingly, *Schneier* does not teach or suggest the feature "writing, to a first record of the second log file, a final second hash value corresponding to a first hash value of the final audit record" as recited in claim 1. Additionally, *Grawrock* does not teach the above-recited features of claim 1, and the Examiner does not assert otherwise.

Grawrock teaches a method for validating a platform within a data center using a cryptographic key pair. *Grawrock* is unrelated and void of any teaching in regards to the authenticity of log files. Thus, *Grawrock* does not teach or suggest the features "responsive to closing the first log file, opening a second log file" and "writing, to a first record of the second log file, a final second hash value corresponding to a first hash value of the final audit record" as recited in claim 1.

Accordingly, because neither *Schneier* nor *Grawrock* teach or suggest all of the features of claim 1, the proposed combination of *Schneier* and *Grawrock* when considered as a whole does not teach or suggest all of the features of claim 1. Therefore, the Examiner fails to state a *prima facie* obviousness rejection of claim 1.

Furthermore, because claims 7 and 10 were rejected for the same reasons as claim 1, the same distinctions between the cited references vis-à-vis claim 1 applies to claims 7 and 10. Similarly, neither *Schneier* nor *Grawrock* teach or suggest all of the features of claim 16.

Consequently, because the remaining claims depend from claims 1, 7, 10, or 16, the proposed combination of *Schneier* and *Grawrock* when considered as a whole does not teach or suggest all of the features of the dependent claims. Thus, under the standards of *In re Royka*, the Examiner fails to state a *prima facie* obviousness rejection of claims 1-20. Therefore, the rejection of claims 1-20 under 35 U.S.C. § 103 has been overcome.

IV.B. No Reason Exists To Combine the References When the References Are Considered as a Whole Because the References Address Different Problems

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). The prior art reference (or references when combined) must teach or suggest all the claim

limitations. *In re Royka,* 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In determining obviousness, the scope and content of the prior art are... determined; differences between the prior art and the claims at issue are... ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or non-obviousness of the subject matter is determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966). Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue. *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. *Id.* (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006)).

For example, *Schneier* is directed to solving the problem of securing audit logs on untrusted machines. In particular, *Schneier* provides that:

> In a system lacking such an ideal communications channel, we nevertheless would like to be able to:
> 1) ensure that an attacker who gains control of U at time t will not be able to read log entries made before time t;
> 2) ensure that the attacker will not be able to undetectably alter or delete log entries, made prior to time t, after U reports its results back to T; and
> 3) provide for occasional "commitments" from U, specifying the current values of its logs in a way that will not allow any later changes to be undetectably made, even if U is compromised.

Schneier, column 2, line 59 – column 3, line4.

On the other hand, *Grawrock is* unrelated to the issue of securing audit logs. However, *Grawrock* is directed to the problem of validating the inclusion of a platform within a data center (*Grawrock*, column 1, lines 11-14). For example, *Grawrock* states:

> Communication networks and the number of users exchanging and transferring data across such networks continue to increase. Specifically, with the advent of the Internet, e-business transactions have caused the exchange and/or transfer of users' personal and/or financial information. Currently, there are little assurances that a user and/or a business attempting to consummate an e-business transaction with a given web site is communicating with the server(s) associated with this web site. In particular, Internet Protocol (IP) snooping is becoming more rampant such that a given computer platform can pretend to be associated with a web site, thereby deceiving users (attempting to perform an e-business transaction with this web site) into disclosing personal and/or financial information.

Grawrock, column 1, lines 7-19.

Based on the plain disclosures of the references themselves, the references address completely distinct problems that are unrelated to each other. The problem of securing audit logs is completely distinct from the problem of validating the inclusion of a platform within a data center.

Because the references address completely distinct problems, when the references are considered together as a whole one of ordinary skill would have no reason to combine or otherwise modify the references to achieve the invention of claim 1. Accordingly, under the standards of *KSR Int'l.*, the Examiner has failed to state a *prima facie* obviousness rejection against claims 1-20.

V. ***Conclusion***

The subject application is patentable over the cited references and should now be in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: August 15, 2007

Respectfully submitted,

/Theodore D. Fay III/

Theodore D. Fay III
Reg. No. 48,504
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777

TF/nh